United States Utility Patent Application

Title: Method and Apparatus for Storing Changes to File Attributes Without Having to Store an Additional Copy of the File Contents

Related Applications

This application claims the benefit of U.S. Provisional Application No. 60/192,244, filed March 22, 2000. U.S. Patent Application No. _______, filed on the same day as this application, and entitled, "Method Of And Apparatus For Recovery Of In-Progress Changes Made In A Software Application," and U.S. Patent Application No. _______, filed on the same day as this application, and entitled, "Method And Apparatus For Automatically Deploying Data In A Computer Network," are hereby incorporated by reference.

Field of the Invention

The invention relates to management of memory resources. More particularly, the invention relates to management of memory resources utilized during development of content for a website, extranet site or intranet site.

Background of the Invention

Data is typically stored in a computer system in the form of data files. A data file may include content and identifying information and may change over time in response to a variety of events. For example, a file may be named, re-named, moved, copied, updated, supplemented, edited or deleted. Thus, the content of a data file or its identifying information may each be altered several times during the course of events. Particularly, for development of a website, extranet site or intranet site, a large quantity of information may need to be stored and may need to be frequently altered. For example, storage may be required for work-in-progress and published editions of the website, extranet site or intranet site.

In computer systems that manage large quantities of data, it is often desirable to keep track of changes to files. This is typically accomplished by storing a prior version of a file and a new version of the file in computer memory each time the file is changed. This storing of file versions can be especially useful in the field of website development where it may often be necessary to revert to a prior version of the website or a portion thereof. A drawback to this technique, however, is that a large amount of physical computer memory can be required to store the various different versions of the files that make up the website.

While the cost of computer memory has generally been decreasing, communication bandwidth has increasingly become a limiting factor in computer systems. Thus, there has been little incentive to reduce the amount of memory required for data storage except to the extent that bandwidth-reducing techniques, such as file compression, also incidentally reduce memory requirements. However, file compression has a disadvantage of requiring additional processing capability to compress and decompress the files. Further, file compression techniques are generally tailored specifically to the type of data contained in the file and are, thus, not universally applicable.

Therefore, what is needed is a technique for minimizing the amount of physical memory required for storing various different versions of changed files. More particularly, what is needed is a technique for minimizing the amount of physical memory required for storing various different versions of a file generated during development of a website, extranet site or intranet site. It is toward these ends that the present invention is directed.

Summary of the Invention

The invention is a method of and an apparatus for storing changes to file attributes without having to store an additional copy of the file contents.

In a computer system that manages a quantity of data, the data is stored in computer memory in form of files. For example, in a system for maintaining and making changes to content for a website, extranet site or intranet site, physical memory may be allocated for work areas, staging areas and edition areas. Work areas may store in-progress changes to be made to the content by an individual contributor or group of contributors. Once the changes are made in the work areas, the changed content may be submitted to a staging area or areas. In the staging area, the content changes may be combined. From the staging area, the changed content may be forwarded to an edition area or areas. The edition areas may store versions or editions of the content for the website, extranet site or intranet site.

In addition to the memory that may be allocated for the various purposes described above, a physical backing store memory may also be provided for providing persistent storage for changes and other content. By persistent, what is meant is that the contents of the backing store are preferably saved despite a shutdown or power failure occurring to the system of which the backing store is a part.

Because the website, extranet site or intranet site may allow access to a large quantity of data, including, for example, text, graphics and audio information, the backing store memory may need to store a large quantity of data. This especially true considering that the backing store may store work-in-progress and multiple editions or versions of content for the website, extranet site or intranet site. For example, the backing store memory may be required store up to three times or more the size of the entire content of the website, extranet site or intranet site.

Files in the backing store or other memory each include contents and attributes. The contents of a file include the information stored within the file. This may include, for example, the text, graphics and audio information which is to be accessible via the website, extranet site or intranet site. The attributes of a file (the attributes may also be referred to as "metadata") include information relating to the file. This may include, for example, the size of the file contents, a time stamp indicating when the file was created or when the file contents were last changed, an identification of an owner who is the contributor responsible for the file, an identification of a group to which the file or its owner belongs and permission information for restricting access to the file for performing read and write operations on the file.

The attributes of a file are stored, such as in the backing store, in association with the contents for the file. When a change is made to the file contents, both the changed version of the file contents and the version prior to the changes may be stored in the backing store. For each version of the file contents, associated attributes are also stored. Because the attributes may include the size of the file and a time stamp indicating when the file was last changed, changes in the file contents generally result in changes to the attributes. Accordingly, a different set of attributes is stored for each version of the file contents.

However, when a change is made to the attributes of the file without a change to the contents of the file, the newly changed attributes may be stored, such as in the backing store, along with the prior version of the attributes. The newly changed attributes and the prior version of the attributes then share the same version of the file contents. In this way, multiple versions of attributes may be associated with a single copy of file contents. Thus, storage space in the backing store is preserved since storage of a separate copy of the file contents in association with each version of the file attributes is avoided. A pointer may be provided, such as in the backing store, which links both the newly changed attributes and the prior attributes to the same copy of the associated file contents.

The present invention is particularly advantageous for reducing the amount of storage space required in a system for development of a website, extranet site or intranet site. This is due, at least in part, to the large quantity of data files generally required to be stored in such systems and the need to store prior versions of files along with new versions of the files.

Brief Description of the Drawings

Figure 1 illustrates a computer network system for website development in accordance with the present invention;

Figure 2 illustrates diagrammatically a work area, a staging area and an edition area, which may be utilized by the system of Figure 1;

Figure 3 illustrates diagrammatically computer memory, such as the backing store memory of Figure 1, for storing files having attributes and contents in accordance with the present invention;

Figure 4 illustrates diagrammatically a file, which is undergoing development stored in the backing store memory of Figures 1 and 3;

Figure 5 illustrates diagrammatically files stored in the backing store memory as a result of making a change to the contents of the file of Figure 4;

Figure 6 illustrates diagrammatically contents of the backing store memory as a result of making a change to the attributes of the file of Figure 4;

Figure 7 illustrates diagrammatically contents of the backing store memory as a result of making another change to the attributes of the file of Figure 6; and

Figure 8 illustrates diagrammatically contents of the backing store memory as a result of making a change to file contents associated with one of the sets of attributes of Figure 7.

Detailed Description of a Preferred Embodiment

Figure 1 illustrates a computer network system 100 for website development. On development workstations 102, website developers may add, remove, edit and examine files for a website. Development workstations 102 may include conventional personal computers, UNIX workstations, or other workstations that can be configured to develop content. The development workstations 102 may be connected to a development server 104 via a computer network 106, such as the Internet or a local area network (LAN).

The development server 104 includes a web server 108 for serving up content to web browsers, and a backing store 110 for storing versions of website content. The server 108 processes hypertext transfer protocol (HTTP) requests from the development stations 102 for website content (e.g., files). The website files may be physically stored in the backing store 110 which may be conventional, such as the WINDOWS NT files system commercially available from Microsoft Corporation.

The development server 104 may also include a conventional memory 112 (e.g., RAM) and a conventional processor 114, which implements the website development methods of the present invention by executing software 116 stored in the memory 112. An HTTP protocol virtualization module 118 which the processor 114 executes to allow web server 108 to operate as if it were multiple servers may also be stored in the memory 112.

The development server 104 may be coupled to a website production web server 120 via a network 122. Network 122 may be the same network as network 106 or a different network. The web server 120 may also be coupled to the Internet or an intranet 124. When a website is ready to be posted on the World Wide Web or on an intranet, the development server 104 sends the website content to the production web server 120 which then provides Internet or intranet access to the website.

A website is generally comprised of the contents of an arbitrary file system. The website development system 100 of the present invention may include hierarchical file systems. Each such file system of the present invention provides an environment for managing and manipulating individual files. When executed, the website development software 116 part of the file system enables the management and manipulation of files. The backing store 110 is where the files and corresponding metadata (e.g., owner identification, group identification, access control, file name, modification times, creation times, etc.) may be physically stored.

A hierarchical file system of the present invention may be referred to as an "area." There may be different types of areas including work areas, staging areas and edition areas. A work area is typically a modifiable file system that is used by persons who create and maintain web content for eventual use in a website.

Memory may be divided into areas utilized for different purposes. For example, Figure 2 illustrates diagrammatically a work area 200, a staging area 202 and an edition area 204. Though a single work area 200, staging area 202 and edition area 204 are illustrated in Figure 2, it will be understood that plural work areas, staging areas and edition areas may be provided. Files 302-316 (Figure 3) may be created or modified in the work area 200. For example, each individual contributor may be assigned a work area, such as the work area 200. Alternately, a group of contributors may be assigned a work area. In addition, the work area 200 may include a physical memory device which is distinct from memory devices utilized for the staging area 202, the edition area 204 or the backing store memory 110 (Figure 1). For example, the work area 200 may include memory of an individual contributor's personal computer system.

The staging area 202 is usually an area where content is assembled before it is published. Since the work areas 200 are generally areas for creating and maintaining content exclusively, the staging area 202 (and the edition area 204), may be restricted to only assembling and displaying content. By design then, the staging 202 and edition 204 areas may be configured as read-only file systems. Any modifications to content may then be sent from an editor back to a workstation to perform any changes that may be needed. Thus, a task for which content is modified may reference content in a staging 202 or edition area 204 but with the modifications actually taking place in a work area 200. This helps to maintain the integrity of the content and to simplify the process. However, a business may want the system 100 to be more flexible, allowing other people such as editors to modify the content directly before it is published. The staging and edition areas may then be configured as modifiable file systems. Thus, in such an embodiment, content submitted from work areas may be edited in a staging area 202 or in an edition area 204.

A work area 200 may initially include a virtual copy of an entire website (unless there is no existing website, in which case the work area may be empty). In other words, a work area 200 may initially have the same contents as the file system designated as the website. A work area 200 provides a developer's personal view of a website in which the developer may contribute to the website. For example, in a work area 200, developers can freely add, delete and modify website content and see how their changes fit into the context of the entire website. Changes made by developers in work areas preferably do not affect the website or the work of other contributors in other work areas. This is because each work area may be a separate file system. Typically, a work area is located at a workstation 102 (Figure 1).

Developers may integrate their work in a staging area by submitting the contents of their work areas 200 into a staging area 202. The staging area 202 is a file system available to multiple developers that provides a shared view of the website. Thus, a staging area 202 may hold the collective work of several developers' work areas 200 and may allow the developers to share and integrate their changes. In a staging area 202, the developers can see how their changes fit together. The staging area is typically located at the development server 104.

Copying is said to be "virtual" where areas share directory trees such that the directory trees do not have to be physically copied. The collective work in a staging area 202 changes as different contributors submit new content from work areas 200. Work areas 200 are most effective when the content and other information in the staging area 202 is virtually copied back to one or more private work areas 200. This helps to keep the individual work areas 200 up-to-date with respect to the staging area 202 while contributors are performing website related tasks such as creating and maintaining content.

When the collective work in a staging area 202 is deemed final, its contents can be published to create an edition of the website. This may be accomplished by virtually copying contents of a staging area 202 onto an edition area 204. Because an edition is typically a read-only file system, it is a frozen snapshot of the content of the entire website. Each edition taken at sequential points in the development of a website may be archived and accessible to all developers so that developers may instantly recall files, entire directories, or reconstruct entire past versions of the website. For example, the contents of an edition may be virtually copied into a work area to be used as a basis for further development of the website. An edition area 204 is typically located in the development server 104. Content (e.g., an edition) in the development server 104 may be deployed to the production server 120.

In sum, once changes are made to a file in the work area 200, the changed file may be submitted to the staging area 202. The staging area 202 may be utilized for combining changes to files made by various contributors in various different work areas 204. Once the changes are combined in the staging area 202, the changed files may be deployed to the edition area 204. The edition area 204 may be utilized for storing versions or editions of the content for the website, extranet site or intranet site.

During this process of making changes, combining and, then, deploying them, the contents of the work area 200, staging area 202 and edition area 204 may be stored in the backing store 110 (Figure 1). For example, once a change made in a work area 200 is submitted to the staging area 202, the change may be stored in the backing store 110.

Figure 3 illustrates diagrammatically a physical computer memory storing files in accordance with the present invention. In a preferred embodiment, the memory serves as the backing store memory 110 in the system 100 (Figure 1) for website, extranet site or intranet development. Referring to Figure 3, the memory 110 stores files 302-316, each file having attributes and contents. Thus, the file 302 includes attributes F01 stored in association with contents F01-1; the file 304 includes attributes F02 stored in association with contents F02-1; the file 306 includes attributes F03 stored in association with contents F03-1; the file 308 includes attributes F04 stored in association with contents F04-1; the file 310 includes attributes F05 stored in association with contents F05-1; the file 312 includes attributes F06 stored in association with attributes F06-1; the file 314 includes attributes F07 stored in association with contents F07-1; and the file 316 includes attributes F08 stored in association with contents F08-1. It will be understood that the memory 110 may form a portion of a conventional computer system or network, including, for example, one or more general purpose processors, software for controlling processor operation, and input/output devices for providing communication among the processors and external devices.

The contents of each of the files 302-316 are the information stored within the file. Because the files 302-316 may be utilized for organizing data which is to be accessible via a website, extranet site or intranet site, the contents may include, text, graphic images, sounds, software scripts, and so forth. The attributes of the files 302-316 include information relating to each file. This may include, for example, the size of the file contents, a time stamp indicating when the file was created or when the file contents were last changed, an identification of an owner who is the contributor responsible for the file, an identification of a group to which the file or its owner belongs and permission information for restricting access to the file for performing read and write operations on the file.

Figure 4 illustrates the file 302 stored in the backing store memory 110. The file 302 of Figure 4 may be undergoing development. Thus, for example, a contributor to the website, extranet site or intranet site of which the file 302 is a part, may make an addition, deletion or change to the contents F01-1 of the file 302. These changes may be made in the work area 200 (Figure 2) and, then, submitted to the staging area 202 (Figure 2).

Figure 5 illustrates diagrammatically files 302, 304 stored in the backing store memory 110 as a result of making a change to the contents F01-1 of the file 302 of Figure 4. The contents F02-1 of the file 304 shown in Figure 5 includes changes, which were made to the contents F01-1 of the file 302. Because the contents F01-1 of the file 302 were changed, the contents F01-1 of the file 302, are stored in the memory 110 along with the original version of the contents F02-1 of the file 304. For example, the file 304 may be created upon submission of the changes to the staging area 202 (Figure 2).

Because the contents F01-1 of the file 302 were changed, this tends to require that the attributes F01 also be altered. For example, one of the attributes F01 (e.g., a size attribute) may indicate the size of the contents F01-1 of the file 302. In which case, a change to the contents F01-1 would likely result in a change to the size attribute. As another example, one of the attributes F01 (e.g., a time stamp attribute) may include a time stamp for the last alteration of the file contents F01-1. In which case, the time at which the contents F01-1 were changed would be reflected by that attribute. Accordingly, the attributes F02 of the file 304 may differ from the attributes F01 of the file 302. Thus, as shown in Figure 5, each set of attributes F01 and F02 may be stored in the backing store memory 110 in association with the corresponding contents F01-1 and F02-1.

Thus, storage of the files 302, 304 in the memory 110 as a result of a change to the contents F01-1 of the file 302 has been described.

Under certain circumstances, however, it may be desired to make a change to the attributes of a file without making any changes to the file contents. For example, the file may be re-assigned to a new owner who is to become the contributor responsible for the file. In which case, the attribute (e.g., an owner attribute) that identifies the owner of the file may be changed without any corresponding change being made to the file contents.

If one or more of the attributes F01 of the file 302 is changed, rather than its contents F01-1, a separate copy of the contents F01-1 need not be stored along with the changed set of attributes. Rather, a single copy of the contents F01-1 may be shared by both the original set of attributes F01 and the changed set of the attributes F02. This is shown in Figure 6, which illustrates diagrammatically contents of the backing store memory 110 as a result of making a change to the attributes F01 of the file 302 of Figure 4.

When a change to the attributes F01 of the file 302 is made, such as in the work area 200 (Figure 2), the change may then be submitted to the staging area 202. As shown in Figure 6, upon submission of the change, a new set of attributes F02, which includes the changes made to the attributes F01, may be stored in the memory 110 along with the original set of attributes F01 and the original contents F01-1 of the file 302. In addition, a pointer 600 may be stored in the backing store memory 110.

The pointer 600 includes an indication that the attributes F02 are associated with the file contents F01-1. For example, when the attributes F02 are formed, an identification of the pointer 600, such as its memory address, may be included in the attributes F02. This associates the pointer 600 with the attributes F02. The pointer 600 may then identify the file contents F01-1. For example, the pointer 600 may include a starting address for the file contents F01-1. This associates the pointer 600 with the file contents F01-1.

As a result of utilizing the pointer 600 to associate the file contents F01-1 with the attributes F02, the file contents F01-1 are shared by both sets of attributes F01 and F02. Accordingly, this conserves space in the backing store memory 110 because there is no need to store a separate copy of the file contents F01-1 for each of the two sets of attributes F01 and F02.

If another change is then made to the file attributes F02 of Figure 6, then a third set of attributes F03 may be formed. Figure 7 illustrates diagrammatically contents of the backing store memory 110 as a result of making another change to the attributes F01 or F02 of the file 302 of Figure 6. For example, if the attributes F02 are changed again, such as to re-assign the file 302 to yet another owner who is to become the contributor responsible for the file 302, then a new set of attributes F03 may be formed. For example, the set of attributes F03 may be formed upon submission of this change to the staging area 202 (Figure 2). An identification of the pointer 600, such as its memory address, may also be included in the attributes F03. This associates the pointer 600 with the attributes F03 in addition to the pointer 600 being associated with the attributes F02. Accordingly, the attributes F01, F02 and F03 may all share the same file contents F01-1.

Rather than including an identification of the pointer 600 in the newly formed attributes F03, a new pointer (not shown) may be formed. In which case, the attributes F03 may include an identification of the newly formed pointer. If the newly formed pointer is to replace the pointer 600, the attributes F02 may also be modified to associate the attributes F02 with the newly formed pointer.

Thus, storage of the files 302 in the memory 110 as a result of a change to the attributes F01 of the file 302 has been described.

If, however, a change is then made to the file contents F01-1, but which change is intended to affect the contents associated with only one of the sets of attributes F01, F02, F03, of Figures 6 or 7, then sharing of the file contents F01-1 may be discontinued. For example, assume that a change or substitution is made to the file contents associated with the attributes F03 of Figure 7, but which is not intended to change the file contents associated with the attributes F01 and F02. For example, the change may be made in the work area 200 (Figure 2) and, then, submitted to the staging area 202 (Figure 2).

As a result, a new file 308 may be formed, as shown in Figure 8. The newly formed file 308 includes attributes F04 and contents F04-1. The attributes F04 may be the same as the attributes F03, assuming no attribute changes were made. However, the attributes F04 may be updated to reflect the changes in the file contents F04-1. For example, a time stamp or file size may need to be altered. The contents F04-1 include the changes made to the file contents F01-1. The attributes F03 may also be retained in the memory 110 in association with the contents F01-1 for the purpose of tracking these changes to the file 302.

In general, the invention may include the utilization of dedicated processors, webservers configured to receive and route browser requests, application servers, state servers and other types of computer processors configured to communicate amongst each other and that may be connected to one or more networks, including a Local Area Network (LAN), an intranet and the Internet. However, it will be appreciated by those skilled in the art, such implementation of such devices and systems are but few illustrations of the utility of the invention, and that the invention may have greater applicability and utility in many other applications where efficient routing and processing of data within one or more networks is involved. Equivalent structures embodying the invention could be configured for such applications without diverting from the spirit and scope of the invention. Although this embodiment is described and illustrated in the context of devices and systems for exchanging data among users of a computer system or network, the invention extends to other applications where similar features are useful. The invention may include personal computers, application servers, state servers or Internet webservers that are designed and implemented on a computer and may be connected to a network for communication with other computers to practice the invention. A system configured to operate according to the invention may include a plurality of personal computers connected to the Internet via individual modems or other communication means such as wireless communications.

The invention may also involve a number of functions to be performed by a computer processor, such as a microprocessor. The microprocessor may be a specialized or dedicated microprocessor that is configured to perform particular tasks by executing machine-readable software code that defines the particular tasks. The microprocessor may also be configured to operate and communicate with other devices such as direct memory access modules, memory storage devices, Internet related hardware, and other devices that relate to the transmission of data in accordance with the invention. The software code may be configured using software formats such as Java, C++, XML (Extensible Mark-up Language) and other languages that may be used to define functions that relate to operations of devices required to carry out the functional operations related to the invention. The code may be written in different forms and styles, many of which are known to those skilled in the art. Different code formats, code configurations, styles and forms of software programs and other means of configuring code to define the operations of a microprocessor in accordance with the invention will not depart from the spirit and scope of the invention, which is defined by the appended claims.

Within the different types of computers, such as computer servers, that utilize the invention, there exist different types of memory devices for storing and retrieving information while performing functions according to the invention. Cache memory devices are often included in such computers for use by the central processing unit as a convenient storage location for information that is frequently stored and retrieved. Similarly, a persistent memory is also frequently used with such computers for maintaining information that is frequently retrieved by a central processing unit, but that is not often altered within the persistent memory, unlike the cache memory. Main memory is also usually included for storing and retrieving larger amounts of information such as data and software applications configured to perform functions according to the invention when executed by the central processing unit. These memory devices may be configured as random access memory (RAM), static random access memory (SRAM), dynamic random access memory (DRAM), flash memory, and other memory storage devices that may be accessed by a central processing unit to store and retrieve information. The invention is not limited to any particular type of memory device, nor any commonly used protocol for storing and retrieving information to and from these memory devices respectively.

While the foregoing has been with reference to particular embodiments of the invention, it will be appreciated by those skilled in the art that changes in these embodiments may be made without departing from the principles and spirit of the invention, the scope of which is defined by the appended claims.